



Installation d'une passerelle

Sous

Debian GNU/Linux

Arnaud Fontaine

(janvier 2004)



Table des matières

1	Introduction	3
2	Installation et sécurisation du système	3
2.1	Recommandations pour et après l'installation	3
2.2	Pam et autres restrictions	4
2.3	Installation d'un noyau	8
2.4	Lilo et le BIOS	9
2.5	Paquets et services inutiles	10
2.6	Mise en place d'un firewall	10
2.7	Mises à jour de sécurité	14
3	Installation et configuration des services	14
3.1	MySQL	15
3.2	Apache	15
3.3	Ssh	17
3.4	Postfix	18
4	Mise en place de prisons chroot	19
4.1	Présentation de Makejail	19
4.2	Configuration d'une prison chroot pour Apache	20
4.3	Configuration d'une prison chroot pour MySQL	22
5	Outils de surveillance du système	24
5.1	Aide	24
5.2	Logcheck	28
6	Conclusion	29
7	Annexe	30
7.1	Arborescence d'une prison chroot pour Apache	30
7.2	Arborescence d'une prison chroot pour MySQL	35

1 Introduction

Ce document décrit l'installation d'une passerelle sous Debian GNU/Linux Woody qui se trouve être la version stable actuellement de cette distribution. Ce document ne se prétend pas exhaustif mais est plutôt destiné à servir de base aux personnes souhaitant mettre en place un système sécurisé sans trop s'embêter. Celui-ci se veut une approche pratique et non théorique, pour cela, je vous conseille de lire la section 'Ressources' qui contient des liens vers d'autres documentations parfois plus théoriques.

En effet, on entend de plus en plus parler du piratage. On remarque que souvent, les failles permettant une intrusion proviennent généralement des personnes qui utilisent ce système, plutôt que le système en lui-même. Cette documentation est donc destinée à vous apprendre pas à pas la configuration d'une passerelle sous Debian GNU/Linux. De plus, il est très important de ne pas se connecter sur Internet avant que le serveur ne soit prêt et que les règles de sécurité de base aient été correctement appliquées.

Pourquoi utiliser cette distribution et pas une autre ? Tout d'abord parce que j'utilise Debian depuis un certain temps, mais aussi parce que cette distribution est réputée pour ses efforts concernant les mises à jour de sécurité, et enfin pour sa gestion performantes des paquets.

2 Installation et sécurisation du système

Il décrit donc en première partie l'installation (sans pour autant rentrer dans les détails) et la sécurisation basique du système. Cette partie donne un aperçu de ce que vous devez absolument faire pour que votre système ne soit pas trop vulnérable.

Je vous conseille fortement de lire cette section même si cela peut vous permettre simple car elle comporte des éléments importants pour la sécurité de base de votre système. En complément de cette section, je vous conseille vivement de lire également le « *Securing Debian Manual* » [1] très complet.

2.1 Recommandations pour et après l'installation

Je ne détaillerai pas l'installation d'une distribution Debian GNU/Linux car cette description a déjà été faite dans un article écrit du guide de l'utilisateur Debian disponible sur Andesi [1]. De plus cela n'est pas vraiment l'objet de ce cours, je vous dirais seulement de n'installer aucun paquets de plus que ceux de l'installation par défaut, et donc de ne pas utiliser *dselect* ou *taskselect*.

Je vais toutefois décrire le partitionnement qui est assez spécifique pour un serveur. Il est vivement recommandé de créer plusieurs partitions. En effet, comme indiqué dans le manuel de Sécurisation de Debian [1] :

- Toute arborescence dans lequel un utilisateur quelconque a des permissions d'écriture tel que */home* ou */tmp* doit être placée dans une partition qui lui est propre.
- Toute arborescence dont la taille varie beaucoup tel que */var* doit aussi être placé sur une autre partition que la racine.
- Il apparaît plus logique de déplacer les données statiques sur une autre partition afin de pouvoir la monter en lecture seule.

Nous choisirons le système de fichier ext3 qui présente l'avantage, par rapport à son prédcesseur, d'être journalisé [6]. Ainsi, en cas de coups dur vous pourrez toujours monter vos partitions en ext2 et éviter les pertes de données. Nous allons affecter quelques options spécifiques aux partitions créées pendant l'installation. Voici un exemple de fichier */etc/fstab* :

Fichier /etc/fstab					
<i># Informations statiques concernant les systèmes de fichier</i>					
<i># <file system></i>	<i><mount point></i>	<i><type></i>	<i><options></i>	<i><dump></i>	<i><pass></i>
/dev/hda5	/	ext3	errors=remount-ro	0	1
/dev/hda6	none	swap	sw	0	0
proc	/proc	proc	defaults	0	0
/dev/hda1	/boot	ext3	defaults,nosuid,ro,nodev,noexec	0	2
/dev/hda7	/home	ext3	rw,nosuid,nodev,noexec,nouser	0	2
/dev/hda8	/usr	ext3	defaults,ro,nodev	0	2
/dev/hda9	/var	ext3	defaults,nodev,nosuid	0	2
/dev/hda10	/tmp	ext3	defaults,nodev,noexec,nosuid	0	2
Fin de /etc/fstab					

Vous pouvez remarquer les options *nosuid*, *noexec*, *ro*, *nodev* qui permettent respectivement le montage de partition sans fichier avec les bits SUID/SGID, sans la permission d'exécuter quoique ce soit, le montage en lecture seule et enfin d'ignorer les périphériques. Notez que ces options ne sont pas synonyme de protection totale car un intrus possédant les droits de l'utilisateur root pourra facilement remonter les partitions comme il le désire, mais cela constitue une bonne protection contre les Script Kiddies.

Cependant, le montage de la partition */usr* en lecture seule peut poser quelques problèmes notamment lors de l'installation ou la mise à jour de paquet, car ces opérations requièrent la possibilité d'écrire sur cette partition. Pour remédier à ce problème, nous allons ajouter un fichier */etc/apt/apt.conf* qui contiendra ceci :

Fichier /etc/apt/apt.conf
<i># On remonte la partition /usr en écriture lors de l'installation ou de la mise à jour de paquets via APT</i>
DPkg
{
Pre-Invoke { "mount /usr -o remount,rw" } ;
Post-Invoke { "mount /usr -o remount,ro" } ;
};
Fin de /etc/apt/apt.conf

Vous devriez aussi activer les mots de passe au format MD5 et shadow, ce dernier étant activé par défaut, afin de rendre vos mots de passe beaucoup plus difficilement crackable. C'est tout ce qu'il était important de ne pas oublier de faire durant l'installation.

2.2 Pam et autres restrictions

Pam (acronyme de « *Pluggable Authentication Modules* ») représente en quelque sorte le système d'authentification de votre système. En effet c'est grâce à celui-ci que vous pouvez vous authentifier quelque soit le programme par lequel vous faites cela. Nous allons décrire brièvement les fichiers de configuration de Pam, une description étant disponible dans le manuel de sécurisation de Debian [1].

Voici quelques fichiers de configuration essentiels de PAM qu'il faut que vous modifiez :

Fichier /etc/pam.d/su
<i>## Le fichier de configuration de PAM pour l'identification des utilisateurs via 'su'</i>
<i># Les modules d'authentification standard d'Unix</i>
auth required pam_unix.so
account required pam_unix.so
Suite . . .

Fichier /etc/pam.d/su (suite)		
session	required	pam_unix.so
<i># On autorise seulement les utilisateurs du groupe wheel à se connecter en root</i>		
auth	required	pam_wheel.so group=wheel debug
Fin de /etc/pam.d/su		

Fichier /etc/pam.d/login		
<i>## Fichier de configuration de PAM pour l'identification des utilisateurs par 'login'</i>		
<i># On empêche à root de s'identifier sur les tty non-listés dans /etc/securetty</i>		
auth	required	pam_securetty.so
<i># On refuse les autres utilisateurs que root à se connecter lorsque /etc/nologin existe</i>		
auth	required	pam_nologin.so
<i># On parcourt ensuite le fichier /etc/environment</i>		
auth	required	pam_env.so
<i># Authentification standard de Unix</i>		
auth	required	pam_unix.so
<i># Compte et sessions standards sous Unix</i>		
account	required	pam_unix.so
session	required	pam_unix.so
<i># Affiche la dernière fois que l'utilisateur s'est authentifié avec succès</i>		
session	optional	pam_lastlog.so
<i># Affiche le contenu du fichier motd dans le cas d'une identification avec succès</i>		
session	optional	pam_motd.so
<i># Affiche le statut des boîtes aux lettres lorsque l'utilisateur s'est authentifié avec succès</i>		
session	optional	pam_mail.so standard noenv
<i># Une méthode plus fiable pour vérifier les mots de passe qui nécessite le paquet 'libpam-cracklib'</i>		
password	required	pam_cracklib.so retry=3 minlen=8 difok=3
password	required	pam_unix.so use_authtok nullok md5
Fin de /etc/pam.d/login		

Fichier /etc/pam.d/passwd		
<i>## Le fichier de configuration de PAM pour 'passwd'</i>		
<i># Le module standard d'authentification, le support des mots de passe au format md5 étant activé</i>		
password	required	pam_unix.so obscure min=4 max=8 md5
Fin de /etc/pam.d/passwd		

Fichier /etc/pam.d/ssh		
<i>## Fichier de configuration de PAM pour l'identification des utilisateurs via 'ssh'</i>		
<i># On refuse les autres utilisateurs que root à se connecter lorsque</i>		
Suite . . .		

Fichier /etc/pam.d/ssh (suite)	
<i>#/etc/nologin existe</i>	
auth	required pam_nologin.so
<i># Authentification standard de Unix</i>	
auth	required pam_unix.so
<i># On parcourt le fichier /etc/environment</i>	
auth	required pam_env.so
<i># On autorise seulement les utilisateurs listés dans</i>	
<i>#/etc/sshusers-allowed à se connecter</i>	
auth	required pam_listfile.so item=user sense=allow file=/etc/sshusers-allowed onerr=fail
<i># Comptes et sessions standards sous Unix</i>	
account	required pam_unix.so
session	required pam_unix.so
<i># On affiche la date de dernière connexion de l'utilisateur</i>	
session	optional pam_lastlog.so
<i># Puis le contenu du fichier motd</i>	
session	optional pam_motd.so
<i># Enfin le contenu de la boîte aux lettres</i>	
session	optional pam_mail.so standard noenv
session	required pam_limits.so
<i># Une autre méthode de vérification des mots de passe plus fiable</i>	
password	required pam_cracklib.so retry=3 minlen=8 difok=3
password	required pam_unix.so use_authtok nullok md5
Fin de /etc/pam.d/ssh	

Fichier /etc/pam.d/other	
<i>## Ce fichier permet de spécifier le comportement de PAM</i>	
<i>## pour les autres applications</i>	
<i># On refuse l'accès par défaut</i>	
auth	required pam_securetty.so
auth	required pam_unix_auth.so
auth	required pam_warn.so
auth	required pam_deny.so
account	required pam_unix_acct.so
account	required pam_warn.so
account	required pam_deny.so
password	required pam_unix_passwd.so
password	required pam_warn.so
password	required pam_deny.so
session	required pam_unix_session.so
session	required pam_warn.so
session	required pam_deny.so
Fin de /etc/pam.d/other	

Une fois que ces fichiers ont été adaptés à votre environnement, il ne faut surtout pas oublier d'ajouter les utilisateurs autorisés à se connecter en root, via la commande `su`, au groupe `wheel`, créez ce dernier si celui-ci n'existe pas déjà par l'intermédiaire des commandes suivantes :

```
# addgroup wheel
# adduser toto wheel
```

Enfin n'oubliez pas non plus d'installer la bibliothèque `libpam-cracklib` sous peine de ne plus pouvoir s'identifier ultérieurement. Je n'ai pas utilisé cette bibliothèque dans le fichier `/etc/pam.d/passwd` car cela posait

quelques problèmes dont je n'ai pas trouvé la solution.

Maintenant que vous en avez terminé avec PAM, quelques petites choses sont encore à configurer. Vous pouvez restreindre les consoles sous lesquelles l'utilisateur root pourra se connecter grâce au fichier décrit ci-dessous :

Fichier /etc/securetty
<i># Liste des terminaux à partir desquels l'utilisateur root est autorisé à s'identifier</i>
console
tty1
tty2
Fin de /etc/securetty

Enfin, on termine cette section en modifiant le fichier `/etc/login.defs` définissant quelques paramètres très utiles et concernant principalement la journalisation ainsi que pour les mots de passe.

Fichier /etc/login.defs
<i>## Paramètres destiné au paquet login</i>
<i># Emplacement de la boîte aux lettres</i>
MAIL_DIR /var/mail
<i># Délai avant que l'identification soit considérée comme échouée</i>
FAIL_DELAY 20
<i># On active la journalisation des information au sujet de /var/log/faillog</i>
FAILLOG_ENAB yes
<i># On active l'affichage des nom d'utilisateurs inconnus</i>
LOG_UNKFAIL_ENAB yes
<i># On désactive la journalisation des authentifications réussites</i>
LOG_OK_LOGINS no
<i># On active les paramètres de ulimit, umask</i>
QUOTAS_ENAB yes
<i># On active la journalisation des 'su'</i>
SYSLOG_SU_ENAB yes
SYSLOG_SG_ENAB yes
<i># Quelques paramètre par défaut</i>
FTMP_FILE /var/log/btmp
SU_NAME su
HUSHLOGIN_FILE .hushlogin
NOLOGIN_STR NOLOGIN
ENV_HZ HZ=100
ENV_SUPATH PATH=/sbin :/bin :/usr/sbin :/usr/bin :/usr/bin/X11 :/usr/local/sbin :/usr/local/bin
ENV_PATH PATH=/usr/local/bin :/usr/bin :/bin :/usr/bin/X11 :/usr/games
<i># Permissions des terminaux</i>
TTYGROUP tty
TTYPERM 0600
<i># D'autres paramètres par défaut...</i>
ERASECHAR 0177
KILLCHAR 025
UMASK 022
<i># On peut ici fixer la durée de validité d'un mot de passe</i>
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_WARN_AGE 7
Suite . . .

Fichier /etc/login.defs (suite)	
<i># Valeurs minimales et maximales des UID lors de l'ajout d'un utilisateur</i>	
UID_MIN	1000
UID_MAX	60000
<i># Valeurs minimales et maximales pour la sélection automatique des GID</i>	
GID_MIN	100
GID_MAX	60000
<i># Nombre maximal de tentatives d'authentification</i>	
LOGIN_RETRIES	5
<i># Durée maximale d'authentification pour 'login'</i>	
LOGIN_TIMEOUT	60
<i># Nombres maximums de caractères pour les mots de passe</i>	
PASS_MAX_LEN	50
<i># Encore des paramètres par défaut</i>	
CHFN_AUTH	yes
CHFN_RESTRICT	rwh
DEFAULT_HOME	yes
USERGROUPS_ENAB	yes
CLOSE_SESSIONS	no
<i># On active le cryptage des mots de passe via md5</i>	
MD5_CRYPT_ENAB	yes
Fin de /etc/login.defs	

2.3 Installation d'un noyau

En général, on installe un serveur sur une machine peu puissante (en tout cas pour les particuliers), vous devez donc économiser au maximum de la mémoire. De plus, vous devriez ne disposer dans votre noyau que des options que vous aurez besoin. C'est pour cela qu'il peut s'avérer utile de compiler un noyau répondant à ces besoins. Par contre je vous conseille fortement de compiler votre noyau à partir d'une autre machine sous Debian GNU/Linux, car il ne faut pas que vous installiez les programmes de développement. Je donnerai dans cette partie seulement les commandes à taper avec peu d'explications car un article plus complet concernant la compilation du noyau à la sauce Debian est disponible sur Andesi [3].

Voilà donc ce que vous devez faire sur une autre machine avec une distribution Debian GNU/Linux préinstallé :

```
# apt-get install build-essential fakeroot kernel-package libncurses5-dev
```

Une fois ces paquets téléchargés et installés, il ne vous reste plus qu'à télécharger la dernière version des sources du noyau sur le site officiel du noyau Linux [4], puis à les décompresser et enfin à compiler le noyau :

```
$ wget http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.24.tar.bz2
$ mkdir ~/src && cd ~/src && tar xvfj ~/linux-2.4.24.tar.bz2
$ wget http://www.andesi.org/articles/autre/passerelle-2.4.24.config \
&& cp ~/passerelle-2.4.24.tex ~/src/linux-2.4.24/.config
$ make menuconfig
$ cd ~/src/linux-2.4.24/ && make-kpkg clean && make-kpkg \
--rootcmd fakeroot --revision=2.4.24-1 kernel-image
```

Enfin copiez l'image du noyau, généré sous la forme d'un paquet Debian, grâce à ssh (en admettant que votre serveur ssh soit actif voir la section concernant ssh) :

```
$ scp ~/src/kernel-image-2.4.23_2.4.24-1.deb toto@192.168.0.1 :~
```

Enfin, revenez sur le serveur puis tapez :

```
# dpkg -i /home/toto/kernel-image-2.4.24_2.4.24-1.deb
```

Vous devez lire la section suivante afin de pouvoir amorcer sur ce nouveau noyau sans aucun problème.

2.4 Lilo et le BIOS

De plus en plus de personnes prennent conscience des risques liés aux attaques distantes, mais il ne faut pas non plus négliger les risques que l'on prend lorsque la machine est accessible physiquement. Une personne mal intentionnée pouvant simplement redémarrer les machine puis démarrer à partir d'un cédérom puis installer un rootkit très facilement. Pour remédier à ce problème, il faut activer l'amorçage à partir du disque dur uniquement et aussi mettre un mot de passe au BIOS afin d'éviter toute modification de ce paramètre par n'importe qui.

Vous pouvez, si vous le souhaitez mettre un mot de passe au gestionnaire d'amorçage qui se trouve être Lilo, installé par défaut sur Debian GNU/Linux. Ainsi, on ne pourra pas démarrer le système en passant par exemple `/bin/bash` comme ligne de commande (cela permet d'arriver directement sur un shell sans avoir à s'identifier auparavant). Néanmoins, en cas de coupure de courant, vous serez obligés d'intervenir pour taper le mot de passe, sans ça votre système ne se lancera pas. Il s'avère plus judicieux de ne me mettre un mot de passe uniquement lorsque des paramètres seront passés à la ligne de commande. Pour cela, on utilise l'option *restricted* et *password*. Un exemple de fichier de configuration de Lilo est fourni ci-dessous :

Fichier /etc/lilo.conf
<pre> # Le support pour les disques dur de haute capacité lba32 # On spécifie le périphérique d'amorçage, c'est à partir de # celui-ci installe son bloc d'amorçage. boot=/dev/hda1 # On spécifie la partition qui doit être montée en tant que root root=/dev/hda5 install=/boot/boot-menu.b map=/boot/map # On empêche de passer des arguments à la ligne de commande en # spécifiant un mot de passe. password=none restricted # On attend 20 secondes avant de lancer le système par défaut delay=20 # Quelques options... prompt timeout=150 vga=normal # Pour passer quelques paramètres à toutes les images # installées append="ether=0,0,eth0 ether=0,0,eth1" default=2.4.18-bf2.4 # Image du noyau par défaut image=/boot/vmlinuz-2.4.18-bf2.4 label=2.4.18-bf2.4 read-only # Et enfin le 2.4.24 fraîchement compilé image=/boot/vmlinuz-2.4.24 label=2.4.24 read-only # N'oubliez pas de lancer 'lilo' une fois ce fichier modifié </pre>
Fin de /etc/lilo.conf

Une fois ce fichier modifié selon vos besoins, il est nécessaire de taper ensuite la commande suivante pour prendre en compte les modification :

```
# lilo
```

2.5 Paquets et services inutiles

Lorsque vous installez un serveur digne de ce nom, tous les paquets et services inutiles doivent être enlevés ou désactivés. Ainsi nous allons désactiver le service inetd sujet à des failles de sécurité (corrigées heureusement mais ce service ne nous est d'aucune utilité). On tape donc la commande suivant pour ne plus démarrer inetd lors de l'amorçage du système :

```
# update-rc.d -f inetd remove
```

Et enfin nous commentons l'ensemble du fichier de configuration de inetd en ajoutant un signe dièse devant chaque ligne dans le cas d'une mise à jour du paquet correspondant. Vous pouvez effectuer la même opération sur l'ensemble de vos services. Pour savoir quels sont les services lancés au démarrage, vous pouvez simplement vérifier le contenu des répertoires `/etc/rc2.d` et `/etc/rcS.d`.

Pour enlever les paquets que vous jugez inutiles pour votre système, vous pouvez tout d'abord vérifier quels sont ceux qui sont installés ainsi que leur description via la commande suivante :

```
# dpkg -l | grep ii
```

Vous pouvez par exemple supprimer telnet que nous remplissons avantageusement par ssh pour des raisons de sécurité. Ensuite supprimez-les (paquets et fichiers de configuration) par l'intermédiaire de la commande suivant :

```
# apt-get remove -purge nom_paquet
```

2.6 Mise en place d'un firewall

La mise en place d'un pare-feu ou firewall est absolument indispensable. Nous allons donc configurer celui-ci avant de se connecter sur internet. La commande en espace utilisateur pour la version 2.4 et 2.6 du noyau GNU/Linux s'appelle Iptables. Netfilter représentant le pare-feu en lui même. Notez que nous ne décrivons pas la configuration d'un pare-feu utilisant ipchains (disponible pour la version 2.2 du noyau).

Dans cette documentation, nous n'allons pas décrire le fonctionnement de Netfilter ainsi que de Iptables car de nombreuses documentations très complètes existent déjà sur le sujet. Nous allons simplement distinguer les commandes qui permettent de manipuler les règles et jeu de règles, les paramètres de spécification des règles et enfin les extensions. Je pense plus utile de fournir un script Iptables fonctionnant correctement plutôt que de décrire à nouveau le fonctionnement de Iptables.

Dans l'exemple suivant, nous considérons que l'interface utilisé pour internet est `ppp0`. J'ai eu l'occasion de tester ce script à plusieurs reprises et j'avoue n'avoir eu aucun problème sur différentes machines. Il se peut cependant qu'il manque quelque chose, dans ce cas n'hésitez pas à me le dire.

Fichier <code>/etc/init.d/iptables.sh</code>
<pre>#!/bin/sh ## Script de pare-feu pour une passerelle proposant un serveur web et ## ssh accessible depuis l'extérieur # Quelques variables à définir avant de pouvoir lancer le script IPT="/sbin/iptables" ext="ppp0" local0="eth0" local1="eth1" lo="lo"</pre>
Suite . . .

Fichier de configuration du pare-feu (suite)

```

# On récupère les adresses IP des DNS du FAI
dns_ip='cat /etc/resolv.conf | grep nameserver | awk -F" " '{print $2}'
## Fonctions de 'nettoyage'
function clean_table ()
{
    # On vide toutes les règles préexistantes
    $IPT -F
    $IPT -X
    $IPT -t nat -F
    $IPT -t nat -X
    # On remet les polices par défaut
    $IPT -P INPUT ACCEPT
    $IPT -P OUTPUT ACCEPT
    $IPT -P FORWARD ACCEPT
}
# Fonction principale définissant les règles à appliquer lors du
## lancement du pare-feu
function start_fw ()
{
    # Pour logger tout ce qui a été rejeté
    $IPT -N LOG_DROP
    $IPT -A LOG_DROP -m limit --limit 1/minute --limit-burst 5 -j \
    LOG --log-level 1 --log-prefix '[IPTABLES DROP] : '
    $IPT -A LOG_DROP -j DROP
    # On autorise tout ce qui sort et venant d'une connexion déjà existante
    $IPT -A INPUT -i $ext -p tcp -m state --state \
    ESTABLISHED,RELATED -j ACCEPT
    $IPT -A OUTPUT -o $ext -p tcp -m state --state \
    ESTABLISHED,RELATED -j ACCEPT
    # Par défaut on rejette tous les paquets
    $IPT -P INPUT DROP
    $IPT -P OUTPUT DROP
    $IPT -P FORWARD DROP
    # On autorise tout ce qui est dans le réseau local
    $IPT -A INPUT -i $local0 -j ACCEPT
    $IPT -A OUTPUT -o $local0 -j ACCEPT
    # Les adresses provenant de classes d'adresses réservées
    $IPT -A FORWARD -p tcp --tcp-flags ALL ALL -j DROP
    $IPT -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP
    # On autorise le trafic en localhost
    $IPT -A INPUT -i $lo -j ACCEPT
    $IPT -A OUTPUT -o $lo -j ACCEPT
    # On autorise le forward via le réseau local
    $IPT -A FORWARD -i $local0 -o $ext -j ACCEPT
    $IPT -A FORWARD -o $local0 -i $ext -j ACCEPT
    # On autorise les requêtes provenant des DNS du FAI
    for i in $dns_ip ; do
        $IPT -A INPUT -i $ext -p udp --sport 53 -s $i -j ACCEPT
        $IPT -A OUTPUT -o $ext -p udp --dport 53 -d $i -j ACCEPT
    done
    # On autorise différents services référencés dans /etc/services

```

Suite . . .

Fichier de configuration du pare-feu (suite)

```

$IPT -A OUTPUT -o $ext -p tcp -m multiport --dports \
ircd,www,pop3,smtp,ssh -m state --state NEW,ESTABLISHED,RELATED -j \
ACCEPT
# On autorise le serveur ssh vers l'extérieur
$IPT -A INPUT -i $ext -p tcp --dport ssh -j ACCEPT
$IPT -A OUTPUT -o $ext -p tcp --sport ssh -j ACCEPT
# On autorise le ftp passif et actif
$IPT -A OUTPUT -p tcp --dport 21 -m state --state \
NEW,ESTABLISHED -j ACCEPT
$IPT -A OUTPUT -p tcp --sport 1 : --dport 1 : -m state --state \
ESTABLISHED,RELATED -j ACCEPT
# On autorise le serveur web vers l'extérieur
$IPT -A INPUT -i $ext -p tcp --dport 80 -m state --state \
NEW,ESTABLISHED,RELATED -j ACCEPT
$IPT -A OUTPUT -o $ext -p tcp --sport 80 -m state --state \
ESTABLISHED,RELATED -j ACCEPT
# On active la passerelle ssi les autres règles sont bien passées
$IPT -t nat -A POSTROUTING -o $ext -j MASQUERADE
# Si on dispose d'un serveur web dans le réseau local
$IPT -t nat -A PREROUTING -i $ext -p udp --dport 8080 -j \
DNAT --to 192.168.0.6 :8080
# Si des paquets ne correspondent pas, on rejete et on journalise
$IPT -A FORWARD -j LOG_DROP
$IPT -A INPUT -j LOG_DROP
$IPT -A OUTPUT -j LOG_DROP
}
## Fonction spécifique au noyau lors du lancement du pare-feu
function kernel_start ()
{
# Quelques options pour le noyau
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
for f in /proc/sys/net/ipv4/conf/*/rp_filter ; do
echo 1 > $f
done
for f in /proc/sys/net/ipv4/conf/*/accept_redirects ; do
echo 0 > $f
done
for f in /proc/sys/net/ipv4/conf/*/send_redirects ; do
echo 0 > $f
done
for f in /proc/sys/net/ipv4/conf/*/accept_source_route ; do
echo 0 > $f
done
for f in /proc/sys/net/ipv4/conf/*/log_martians ; do
echo 1 > $f
done
}

```

Suite . . .

Fichier de configuration du pare-feu (suite)

```

}
## Fonction spécifique au noyau lors de l'arrêt du pare-feu
function kernel_stop ()
{
    # Quelques options pour le noyau
    echo 0 > /proc/sys/net/ipv4/ip_forward
    echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
    echo 0 > /proc/sys/net/ipv4/tcp_syncookies
    echo 0 > /proc/sys/net/ipv4/conf/all/log_martians
    echo 0 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
    echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
    for f in /proc/sys/net/ipv4/conf/*/rp_filter ; do
        echo 0 > $f
    done
    for f in /proc/sys/net/ipv4/conf/*/accept_redirects ; do
        echo 1 > $f
    done
    for f in /proc/sys/net/ipv4/conf/*/send_redirects ; do
        echo 1 > $f
    done
    for f in /proc/sys/net/ipv4/conf/*/accept_source_route ; do
        echo 1 > $f
    done
    for f in /proc/sys/net/ipv4/conf/*/log_martians ; do
        echo 0 > $f
    done
}
## Enfin le démarrage du script proprement dit
case $1 in
    # Lancement du pare-feu !
    start)
        echo -n "Starting Firewall rules"
        clean_table &&
        start_fw &&
        kernel_start
        # Lancement de la connexion aussi
        /usr/bin/pon dsl-provider > /dev/null 2>&1
        echo "."
        ;;
    # Arrêt du pare-feu !
    stop)
        echo -n "Cleaning Firewall table"
        clean_table &&
        kernel_stop
        # On arrête aussi la connexion internet
        /usr/bin/poff -a > /dev/null 2>y&1
        echo "."
        ;;
    # Si on relances la configuration, à utiliser lors de toute modification de ce fichier !
    restart)
        echo -n "Restarting Firewall rules"

```

Suite . . .

Fichier de configuration du pare-feu (suite)	
<pre> clean_table && start_fw && kernel_start echo "." ;;) echo "Usage : /etc/init.d/iptables.sh {start stop restart}" exit 1 ;; esac </pre>	
Fin de /etc/init.d/iptables.sh	

Une fois ce script adapté éventuellement à vos besoins et que celui-ci a été copié dans le répertoire `/etc/init.d/`, il ne vous reste plus qu'à le rendre exécutable puis à créer les liens pour que le pare-feu soit lancé à chaque démarrage de la machine, ceci pourra se faire simplement sous Debian via la commande `update-rc.d` :

```
# chmod 755 /etc/init.d/iptables.sh
# update-rc.d iptables.sh defaults 20
```

Vous pourrez trouver plus d'informations au sujet de la gestion des services lancés au démarrage dans la documentation disponible sur Andesi [2].

2.7 Mises à jour de sécurité

Si vous avez installé Debian GNU/Linux depuis les cédéroms, il faut absolument mettre à jour les paquets de votre système afin de corriger des failles éventuelles. Cela se fait très simplement en spécifiant les sources APT dans le fichier adapté. Voici un exemple de ce fichier :

Fichier /etc/apt/sources.list
<pre> ## <i>Miroir officiel pour stable</i> deb http://http.us.debian.org/debian stable main contrib non-free ## <i>Mises à jour de sécurité</i> deb http://security.debian.org/ stable/updates main </pre>
Fin de /etc/apt/sources.list

Une fois que vous avez modifié ce fichier, vous devrez taper les commandes suivantes pour mettre à jour respectivement la liste des paquets et d'installer les mises à jour disponibles :

```
# apt-get update
# apt-get upgrade -u
```

Normalement peu de paquets seront mis à jour lors de cette étape. Enfin tout dépend évidemment des paquets que vous avez installé. Enfin vous pouvez vérifier les paquets affectés par une faille de sécurité et corrigés sur la partie sécurité du site de Debian [5]. Vous pouvez ensuite installer quelques paquets utiles :

```
# apt-get install locales libpam-cracklib
```

3 Installation et configuration des services

Cette partie concerne la mise en place de services disponibles aux utilisateurs du réseau local et également via internet. Elle décrit donc l'installation d'un serveur web *Apache*, d'un serveur de base de données *MySQL*, d'un service permettant un accès sécurisé à distance *SSH*. Ces trois services étant disponible par Internet. Les autres services sont destinés aux utilisateurs du système ou du réseau local. Avec notamment un serveur de mail *Postfix*, d'un proxy *Squid* et de *Rsync* qui vous permettra de faire des sauvegardes du système.

3.1 MySQL

MySQL est un serveur de bases de données libre, performant et très utilisé. De plus, celui-ci peut s'utiliser sans aucun problème avec le module PHP de Apache. Celui-ci est utilisé sur énormément de serveurs actuellement.

Nous allons donc décrire la configuration minimale à faire pour configurer un tel serveur. Cela est relativement simple si on connaît déjà un peu quelques commandes SQL. Nous installons donc les paquets correspondants :

```
# apt-get install mysql-client mysql-common mysql-server libmysqlclient
```

Une fois cette opération effectuée, le plus dur a été fait. Il ne vous reste plus qu'à décommenter la ligne comportant l'option `skip-networking`. Ensuite redémarrez le serveur MySQL par la commande :

```
# /etc/init.d/mysql restart
```

On peut dire que la configuration de MySQL est terminée, néanmoins, il faut mettre un mot de passe pour l'utilisateur root de la base de données (celui-ci étant totalement différent de l'utilisateur root du système). Pour cela on tape la commande :

```
# mysqladmin -u root password your_password
```

Maintenant on peut se connecter à la base de données avec le mot de passe précédemment assignés :

```
# mysql -u root -p
```

Créons ensuite une base de données ainsi qu'un utilisateur ayant tous les droits dessus :

```
> CREATE DATABASE toto ;
> USE mysql ;
> GRANT ALL PRIVILEGES on toto.* to toto@localhost IDENTIFIED BY 'your_pass' ;
```

Désormais l'utilisateur toto pourra se connecter à sa base de données en tapant la même commande que précédemment :

```
$ mysql -u toto -p
> USE toto ;
```

La configuration du serveur MySQL est maintenant terminée. Cela n'est vraiment pas compliqué, pour une introduction à MySQL, je vous conseille le manuel officiel très complet et disponible en français.

3.2 Apache

Apache est un serveur web très populaire, son succès est du tout d'abord à sa liberté d'utilisation mais également parce qu'il est puissant et stable à la fois. Je vous conseille donc fortement de choisir Apache pour votre serveur. De plus il est disponible en paquet Debian. Nous allons également installer PHP en tant que module pour Apache afin de pouvoir disposer de ce langage de programmation web de plus en utilisé pour la réalisation de sites internet dynamiques. Je ne vais pas décrire la configuration d'apache dans cette section mais plutôt les modifications à faire au niveau du fichier de configuration.

Commençons tout d'abord par installer le serveur Apache ainsi que le module PHP en tapant la commande suivante :

```
# apt-get install apache apache-common php4 php4-mysql
# apacheconfig
```

Le dernier des paquets est destiné à permettre l'utilisation des fonctions MySQL dans PHP. Enfin la dernière des commandes est destiné à relancer la détection des modules installés. Une fois ces paquets installés, il vous reste seulement à modifier les fichiers de configuration de Apache et celui de PHP éventuellement. La configuration que je vais décrire est adapté pour un serveur web avec un minimum de sécurité mais pour un hébergeur, il faudrait que celle-ci soit beaucoup plus stricte.

On modifie d'abord le fichier de configuration de Apache en ajoutant si ce n'est pas déjà fait les extensions des scripts au format PHP et quelques autres options dont la description est donné dans les commentaires.

Fichier /etc/apache/httpd.conf	
	<i># Fichier de configuration partiel d'apache avec simplement les # numéros de lignes des options à modifier avec ce que ça doit donner # Adresse sur laquelle le serveur écoute</i>
L183	Listen aaa.bbb.ccc.ddd :80
	<i># Si on a pas besoin de scripts cgi</i>
L215	# LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
	<i># Pour le support du PHP et sa sécurité</i>
L241	LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
L290	php_admin_flag engine on
L291	php_admin_value open_basedir "/var/www"
	<i># Quelques informations pour vous contacter</i>
L296	ServerAdmin webmaster@host.com
L310	ServerName host.com
	<i># Pour les scripts en PHP</i>
L757	AddType application/x-httpd-php .php .php4
L758	AddType application/x-httpd-php-source .phps
Fin de /etc/apache/httpd.conf	

Une fois que la configuration a été adapté selon vos paramètre, vous n'avez plus qu'à relancer Apache :

```
# /etc/init.d/apache restart
```

Maintenant que nous avons correctement configuré Apache, il ne reste plus qu'à modifier quelques détails dans le fichier de configuration de php afin de renforcer un peu la sécurité de celui-ci. Voici encore un exemple de fichier de configuration avec seulement les lignes que j'ai modifié. Une fois que vous avez modifié ce fichier avec les valeurs indiquées, il faudra redémarrer le serveur Apache afin que les modifications soient prises en compte.

Fichier /etc/php4/apache/php.ini	
	<i># Fichier de configuration partiel de PHP avec simplement les # numéros de lignes des options à modifier avec ce que ça doit donner # Quelques options de sécurité pour PHP</i>
L124	safe_mode = On
L129	safe_mode_gid = On
L139	safe_mode_exec_dir = /var/www
L162	disable_functions = move_uploaded_file,symlink,lstat,stat, \ virtual,bzopen,dba_open,pspell_config_personal,pspell_config_repl, \ curl_init,ini_set,ini_alter,ini_restore,sleep,usleep,syslog, \ posix_kill,fsockopen,pfsockopen,ftp_connect
L722	session.use_trans_sid = 1
Fin de /etc/php4/apache/php.ini	

Vous pouvez à tout moment vérifier que votre serveur tourne correctement en tapant la commande 'ps'. Si ce n'est pas le cas, vérifiez dans les journaux de Apache qui se trouve dans /var/log/apache/, le fichier error.log de ce répertoire étant le plus important dans le cas où votre serveur web rencontre un problème quelconque.

3.3 Ssh

Ssh est un remplacement très intéressant pour Telnet car il présente l'énorme avantage de crypter la communication de bout en bout. Il vous permettra donc d'accéder à distance à votre machine afin de l'administrer, vous économisant ainsi un écran et autres fioritures. De plus la configuration d'un tel serveur est très simple à faire.

Il faut tout d'abord installer le paquet de OpenSSH disponible bien entendu en tant que paquet Debian :

```
# apt-get install ssh
```

Enfin modifiez le fichier de configuration de OpenSSH dont la syntaxe est vraiment très simple à comprendre :

Fichier /etc/ssh/sshd_config
<pre><i># Port et adresse sur lequel le serveur ssh écoute</i> Port 22 ListenAddress aaa.bbb.ccc.ddd <i># On utilise la version du protocole qui est beaucoup plus fiable</i> Protocol 2 <i># L'emplacement où sont stockés les clés privés du serveur</i> HostKey /etc/ssh/ssh_host_rsa_key HostKey /etc/ssh/ssh_host_dsa_key <i># Lors de la connexion, un processus enfant est exécutée, après</i> <i># une authentification réussie, un processus est crée avec les</i> <i># droits de l'utilisateur</i> UsePrivilegeSeparation yes <i># On autorise seulement l'utilisateur 'toto' à se connecter et le groupe 'users', optionnel</i> AllowUsers arno AllowGroups users <i># Déconnexion au bout de 120s si l'utilisateur ne s'est pas authentifiée avec succès</i> LoginGraceTime 120 <i># Nombre maximum de connexions pas encore authentifiée en même temps</i> MaxStartups 1 <i># Journalisation</i> SyslogFacility AUTH LogLevel INFO <i># On utilise pas 'root' à se connecter puis on vérifie les permissions de l'utilisateur</i> PermitRootLogin no StrictModes yes <i># Différentes formes d'authentification...</i> RSAAuthentication no PubkeyAuthentication yes RhostsAuthentication no PAMAuthenticationViaKbdInt no IgnoreRhosts yes RhostsRSAAuthentication no HostbasedAuthentication no PermitEmptyPasswords no PasswordAuthentication yes <i># On refuse le forwarding de sessions graphiques</i> X11Forwarding no X11DisplayOffset 10 <i># Enfin on affiche le fichier motd et la date de dernière connexion</i> PrintMotd no PrintLastLog yes KeepAlive yes</pre>
Fin de /etc/ssh/sshd_config

Vous avez sans doute remarquer que nous n'avons pas autorisé l'utilisateur root à se connecter, tout simplement car cela forcera l'utilisateur qui a les permissions nécessaires à taper deux mots de passe consécutivement. Ainsi on évite les attaques par force brute contre le compte root.

Nous avons choisi d'accepter deux formes d'authentification, la première est classique (défini par l'option `PasswordAuthentication`) car elle nécessite simplement que vous tapiez le mot de passe du compte sur lequel vous souhaitez vous connecter. La seconde forme (défini par l'option `PubkeyAuthentication`) est plus intéressante, selon moi, car elle permet la connexion par clé publique. Ainsi, si un utilisateur souhaite se connecter sur la machine qui accepte uniquement cette forme, il devra obligatoirement disposer d'un mot de passe pour déverrouiller sa clé privée, qui aura été préalablement ajoutée sur le serveur, afin de se connecter.

Voyons comment utiliser cette seconde méthode. Nous allons tout d'abord créer une clé publique et privée sur la machine cliente par la commande suivante :

```
# ssh-keygen -t dsa
```

Une fois le mot de passe choisi et cette clé générée, il ne vous reste plus qu'à copier la clé publique de votre client sur le serveur dans le fichier `~/ .ssh/authorized_keys` de ce dernier, à l'aide la commande :

```
ssh-copy-id -i ~/.ssh/id_dsa.pub toto@ip_serveur
```

Désormais vous pourrez vous connecter sur le serveur en utilisant la méthode d'authentification par clé publique. Un pirate éventuel devra disposer à la fois du mot de passe de la clé mais aussi de l'assortiment de clés proprement dites.

Un serveur ssh pourra remplacer avantageusement un serveur FTP si peu d'utilisateurs utilisent cette machine, car vous pourrez utiliser Gftp comme client pour vous connecter sur votre serveur ssh. Celui-ci se comportera exactement comme un serveur FTP classique.

3.4 Postfix

Postfix est un serveur de mail plutôt connu qui est censé être plus sécurisé que Sendmail et dont la configuration est plus aisée que ce dernier. Celui-ci va nous servir de serveur de mail local et va nous permettre également d'envoyer des courriels en se servant d'un serveur SMTP (celui de votre FAI par exemple).

Nous allons commencer, comme d'habitude, par installer ce paquet :

```
# apt-get install postfix
```

Répondez aux différents choix qui vous sont proposés, vous devriez obtenir un fichier de configuration de postfix qui ressemble à celui-ci :

Fichier <code>/etc/postfix/main.cf</code>
<pre># Quelques répertoires importants pour le fonctionnement de Postfix command_directory = /usr/sbin daemon_directory = /usr/lib/postfix program_directory = /usr/lib/postfix # Quelques paramètres... smtpd_banner = \$myhostname ESMTP \$mail_name (Debian/GNU) setgid_group = postdrop biff = no append_dot_mydomain = no myhostname = your_host alias_maps = hash :/etc/aliases alias_database = hash :/etc/aliases</pre>
Suite . . .

Fichier /etc/postfix/main.cf (suite)
<pre>myorigin = /etc/mailname mydestination = localhost.localdomain, localhost, your_host relayhost = smtp.free.fr mynetworks = 127.0.0.0/8 inet_interfaces = localhost mailbox_command = procmail -a "\$EXTENSION" mailbox_size_limit = 0 recipient_delimiter = +</pre>
Fin de /etc/postfix/main.cf

Enfin voici le contenu du fichier destiné à rediriger les courriels envoyés à l'utilisateur root vers un autre utilisateur. Ceci n'est pas obligatoire bien entendu.

Fichier /etc/aliases
<pre><i># Tous les courriels que reçoit root sont automatiquement</i> <i># redirigés vers la boîte aux lettre de toto</i> postmaster : root root : toto webmaster : root</pre>
Fin de /etc/aliases

Une fois la configuration terminée, vous pouvez redémarrer le serveur de courriels, afin que les modifications soient prises en compte, grâce à la commande suivante :

```
# /etc/init.d/postfix restart
```

4 Mise en place de prisons chroot

Dans cette partie, nous décrivons la mise en place de prisons chroot pour différents services clé que sont Apache et MySQL. La configuration de ces services que vous avez fait dans les précédentes sections ne seront bien sûr pas inutiles puisqu'on va se baser sur exactement les mêmes fichiers de configuration.

Vous vous demandez certainement ce qu'est une prison chroot ? Admettons que nous laissons le système configuré ainsi, une faille sur MySQL pourrait permettre à un attaquant d'accéder au système sous l'utilisateur du processus MySQL. Même si celui-ci n'a pas beaucoup de droits, il serait alors possible d'exploiter une faille locale permettant d'accéder au compte root, ce qui peut se révéler catastrophique car l'ensemble de l'arborescence du système serait touché. Pour résoudre ce problème, il existe un mécanisme permettant d'enfermer le processus dans une arborescence minimale, qui serait donc à l'intérieur de l'arborescence du système. Ce mécanisme s'appelle chroot. On installera dans cette prison seulement les fichiers nécessaires au service, ainsi l'attaquant se retrouvera en théorie bloqué dans l'arborescence et ne pourra donc pas corrompre le reste du système.

4.1 Présentation de Makejail

Vous pourriez très bien copier manuellement les fichiers nécessaires aux services, mais cela s'avère plutôt fastidieux. Makejail vient donc à notre rescousse en permettant de copier automatiquement les fichiers nécessaires au bon fonctionnement du système. Cet utilitaire est écrit en Python et permet de faire des mises à jour de la prison chroot beaucoup plus facilement qu'avec l'ancienne méthode.

Makejail n'est malheureusement pas disponible en tant que paquets officiels pour la version stable de Debian. Cependant j'ai fait un rétro-portage de ce paquet afin qu'il soit utilisable sur woody. Pour installer ce paquet, il vous suffit de taper les commandes :

```
# echo "deb http://www.andesi.org/public/ stable main" > /etc/apt/sources.list
```

```
# apt-get update
# apt-get install makejail stat strace binstats python
```

Il faut savoir que vous pouvez très bien créer des prisons chroot pour d'autres services tels que Ssh ou Bind, dont la construction n'est pas décrite dans cet article. Passons donc à la création d'une prison chroot pour Apache et MySQL, même si cela peut paraître plutôt compliqué, en fait il n'en est rien.

4.2 Configuration d'une prison chroot pour Apache

La création de cette prison chroot nécessite auparavant la création d'un fichier de configuration pour Makejail écrit en Python plutôt facile à faire. En voici un exemple :

Fichier /etc/makejail/apache.py
<pre><i># Emplacement de la prison</i> chroot="/var/chroot/apache" <i># Commande destinée à lancer le service afin de vérifier les</i> <i># fichiers nécessaires au fonctionnement de celui-ci</i> testCommandsInsideJail=["/usr/sbin/apachectl start"] <i># Nom du processus à 'tracer'</i> processNames=["apache"] <i># Une commande destiné à vérifier le bon fonctionnement du service</i> testCommandsOutsideJail=["wget -r -spider http://localhost/"] <i># Les fichiers et répertoires à ne pas copier dans la prison</i> preserve=["/var/www", "/var/log/apache", "/dev/log"] <i># Utilisateur sous lequel tourne le processus</i> users=["www-data"] groups=["www-data"]</pre>
Fin de /etc/makejail/apache.py

N'oubliez pas de créer le répertoire `/etc/makejail/` s'il n'existe pas déjà. Maintenant que vous avez créé ce fichier, vous pouvez créer le répertoire qui va accueillir la prison chroot pour Apache, on arrête le service Apache, puis on lance makejail qui effectue les opérations nécessaires :

```
# mkdir -p /var/chroot/apache/
# /etc/init.d/apache stop
# makejail /etc/makejail/apache.py
```

Il peut être nécessaire de lancer Makejail deux fois afin que la totalité des fichiers soient copiés. Vous pouvez tester dès maintenant le fonctionnement du service en montant le système de fichier virtuel `/proc` dans la prison puis en lançant le service :

```
# mount -t proc proc /var/chroot/apache/proc
# chroot /var/chroot/apache/ /usr/sbin/apachectl start
```

Normalement vous ne devriez avoir aucune erreur à ce stade. Copiez ensuite, le répertoire publique de votre serveur web actuel ainsi que les journaux :

```
# cp -a /var/www /var/chroot/apache/var/
# mkdir /var/chroot/apache/var/log && chmod 755 /var/chroot/apache/var/log/
# cp -a /var/log/apache /var/chroot/apache/var/log/
```

Configurons désormais pour que le service soit automatiquement lancé dans la prison chroot. Pour cela il faut modifier le fichier présent dans `/etc/init.d/` qui permet le lancement du service en toute simplicité. Cela vous évitera donc de taper de longues commandes pour démarrer votre serveur web.

Fichier /etc/init.d/apache
<pre>#!/bin/bash # Quelques variables utiles pour le reste du script CHROOT=/var/chroot/apache NAME=apache PATH=/bin :/usr/bin :/sbin :/usr/sbin DAEMON=/usr/sbin/apache SUEXEC=/usr/lib/apache/suexec PIDFILE=\$CHROOT/var/run/\$NAME.pid CONF=/etc/apache/httpd.conf APACHECTL=/usr/sbin/apachectl trap "" 1 export LANG=C export PATH # Démarrage du script proprement dit case "\$1" in # Pour le démarrage du serveur start) echo -n "Starting web server : \$NAME" mount -t proc proc \$CHROOT/proc > /dev/null 2>&1 & chroot \$CHROOT /usr/sbin/apachectl start > /dev/null 2>&1 echo "." ;; # Pour l'arrêt du serveur stop) echo -n "Stopping web server : \$NAME" umount \$CHROOT/proc > /dev/null 2>&1 & killall apache > /dev/null 2>&1 & echo "." ;; # Si on ne passe aucune option *) echo "Usage : /etc/init.d/\$NAME {start stop}" exit 1 ;; esac</pre>
Fin de /etc/init.d/apache

Maintenant, Apache se lancera sans aucun problème à chaque démarrage de votre serveur dans la prison chroot, même si ce script est loin d'être parfait, il fonctionne plutôt bien chez moi. Modifiez enfin le service syslogd afin qu'il 'capture' tous les messages qui passe dans la chroot de Apache, voici ce que vous devriez obtenir :

```
SYSLOGD=" -a /var/chroot/apache/dev/log"
```

Modifions enfin le fichier de configuration de LogRotate afin de ne pas disposer de journaux trop lourd dans la prison chroot pour apache :

Fichier /etc/logrotate.d/apache
<pre># Pour la rotation des journaux dans la prison chroot pour Apache /var/chroot/apache/var/log/apache/*.log { weekly missingok</pre>
Suite . . .

Fichier de /etc/logrotate.d/apache (suite)
<pre> rotate 52 compress delaycompress notifempty create 640 root adm sharedscripts postrotate /etc/init.d/apache reload > /dev/null endscript } </pre>
Fin de /etc/logrotate.d/apache

Vous pouvez vérifier le bon fonctionnement de Apache directement dans les journaux de la prison chroot qui sont les fichiers du répertoire `/var/chroot/apache/var/log/apache`. La prison chroot pour Apache est maintenant complètement terminée, nous allons maintenant aborder la création d'une prison chroot pour MySQL qui n'est pas plus difficile que celle-ci.

4.3 Configuration d'une prison chroot pour MySQL

On va procéder exactement de la même façon que pour la prison chroot pour Apache. On commence donc par créer un fichier de configuration pour makejail destiné à MySQL dont je donne un exemple ci-dessous :

Fichier /etc/makejail/mysqld.py
<pre> # Emplacement de la prison chroot="/var/chroot/mysqld" # Nom du service à lancer dans la prison chroot testCommandsInsideJail=["safe_mysqld"] sleepAfterStartCommand=5 # Nom du processus à 'tracer' processNames=["mysqld"] # Commande pour tester la prison chroot une fois la prison construite testCommandsOutsideJail=["mysqld -user=root -socket='/var/chroot/mysqld/var/run/mysqld/mysqld.sock' -exec='show t preserve=["/var/lib/mysql", "/var/log/mysql"] # Utilisateur et groupe sous lesquels tourne le service MySQL users=["mysql"] groups=["mysql"] </pre>
Fin de /etc/makejail/mysqld.py

On crée le répertoire qui dans lequel on va placer la prison chroot pour MySQL, puis on arrête le service et enfin on lance makejail qui va s'occuper de copier pour nous tous les fichiers et répertoires nécessaires à MySQL :

```

# mkdir /var/chroot/mysqld/
# /etc/init/mysql stop
# makejail /etc/makejail/mysqld.py

```

Comme pour le cas de Apache, il peut être nécessaire de lancer deux fois makejail afin que tous les fichiers et répertoires soient copiés dans la prison chroot pour MySQL. Copions les répertoires qui ont été préservés par Makejail :

```

# cp -a /var/lib/mysql /var/chroot/mysqld/var/lib/
# mkdir /var/chroot/mysqld/var/log/
# cp -a /var/log/mysql/ /var/chroot/mysqld/var/log/

```

Testons enfin le bon fonctionnement de la prison chroot en montant tout d'abord le système de fichiers virtuel /proc puis on tente de lancer le service :

```
# mount -t proc proc /var/chroot/mysqld/proc/
# chroot /var/chroot/mysqld/ /usr/bin/safe_mysqld &
```

Si tout va bien, le serveur MySQL est démarré dans la prison chroot, on peut aisément vérifier cela avec la commande `ps`. Enfin on modifie le script permettant le lancement du service afin que ce dernier démarre dans la prison chroot qui lui est réservée.

Fichier /etc/init.d/mysql
<pre>#!/bin/sh -e # Quelques variables utiles CHROOT=/var/chroot/mysqld SELF=\$(cd \$(dirname \$0) ; pwd -P)/\$(basename \$0) cd / umask 077 export PATH=/bin :/usr/bin :/usr/sbin # Enfin la partie permettant de démarrer le serveur proprement dit case "\$1" in # Pour démarrer le service 'start') echo -n "Starting MySQL database server : mysqld" mount -t proc proc \$CHROOT/proc > /dev/null 2>&1 & chroot \$CHROOT /usr/bin/safe_mysqld > /dev/null 2>&1 & echo "." ;; # Pour l'arrêt du service 'stop') echo -n "Stopping MySQL database server : mysqld" umount \$CHROOT/proc > /dev/null 2>&1 & killall -15 mysqld > /dev/null 2>&1 & sleep 1 killall -9 mysqld > /dev/null 2>&1 & echo "." ;; # Si aucun argument *) echo "Usage : \$SELF start stop" exit 1 ;; esac</pre>
Fin de /etc/init.d/mysql

Voilà, grâce à ce script, qui, il faut bien l'avouer est loin d'être parfait mais qui a l'avantage de fonctionner, vous n'aurez pas à taper de longues lignes de commandes pour démarrer votre serveur MySQL. Il faut aussi modifier le script de démarrage du service `sysklogd` afin qu'il puisse capter les messages émis par MySQL, vous devriez donc avoir cela dans `/etc/init.d/sysklogd` :

```
SYSLOGD="-a /var/chroot/apache/dev/log -a /var/chroot/mysqld/dev/log"
```

Modifions également le fichier de configuration de LogRotate afin que les journaux dans la prison chroot ne deviennent pas trop importants :

Fichier /etc/logrotate.d/mysql-server
<pre># Pour la rotation des journaux dans la prison chroot pour MySQL /var/chroot/mysqld/var/log/mysql.log \ /var/chroot/mysqld/var/log/mysql/mysql.log \ /var/chroot/mysqld/var/log/mysql.err \ /var/chroot/mysqld/var/log/mysql/mysql.err { daily rotate 7 missingok create 600 mysql mysql compress sharedscripts postrotate MYADMIN="chroot /var/chroot/mysqld /usr/bin/mysqladmin \ -defaults-extra-file=/etc/mysql/debian.cnf" if [-n "\$MYADMIN ping 2>/dev/null"]; then \$MYADMIN flush-logs fi errlogs='ls /var/chroot/mysqld/var/log/mysql.err* \ /var/chroot/mysqld/var/log/mysql/mysql.err* 2>/dev/null' if [-n "\$errlogs"]; then chown 0.0 \$errlogs fi endscrip } </pre>
Fin de /etc/logrotate.d/mysql-server

Un petit détail avant de terminer cette partie sur la mise en place d'une prison chroot pour MySQL, vous devrez préciser dans vos script, qui devront pouvoir se connecter au serveur MySQL, le serveur suivant : 12.0.0.1 :3306.

5 Outils de surveillance du système

Cette section concerne la mise en place d'outils de surveillance (on parle aussi de monitoring), tel qu'un outil de vérification de l'intégrité du système, on utilisera pour cela *Aide*. Enfin on décrira la configuration d'un autre outil (*Logcheck*) permettant de vérifier les évènements suspects dans les journaux du système.

5.1 Aide

AIDE [7](acronyme de « *Advanced Intrusion Detection Environment* ») permet la création d'une base de données des fichiers et répertoires spécifiés dans le fichier de configuration `/etc/aide/aide.conf`. L'intérêt de cette base de données est qu'elle contient les attributs des fichiers ainsi qu'une somme de contrôle de ceux-ci vous permettant de savoir si des fichiers ont été modifiés sur votre système après une attaque par exemple.

Il convient ainsi de créer cette base de données avant de se connecter à un réseau quelconque afin de créer une base de données saine et de ne pas ajouter dans la base de données des fichiers qui sont très souvent modifiés afin d'obtenir un rapport court et simple à examiner.

Vous pouvez installer très facilement AIDE car il est disponible en paquet Debian pour Woody, on tape donc la commande :

```
# apt-get install aide
```

Lors de l'installation du paquet, vous pourrez créer la base de données directement. Une fois ce paquet installé

il ne vous reste plus qu'à adapter le fichier de configuration en suivant par exemple l'exemple donné ci-dessous. Ce fichier est celui par défaut du paquet auquel j'ai ajouté des options pour les environnements chroot pour *Apache* et *MySQL*.

Fichier /etc/aide/aide.conf
<pre> # On définit tout d'abord l'emplacement de la base de données database=file :/var/lib/aide/aide.db database_out=file :/var/lib/aide/aide.db.new # On compresse la base de données au format gzip # Changer cette option à "no" pour un vieux processeur gzip_dbout=yes # Voici tout ce que nous pouvons vérifier - règles par défaut # #p : permissions #i : inode #n : nombre de liens #u : utilisateur #g : groupe #s : taille #b : nombre de blocs #m : mtime #a : atime #c : ctime #S : vérification pour une taille plus importante #md5 : somme de contrôle md5 #sha1 : somme de contrôle sha1 #rmd160 : somme de contrôle rmd160 #tiger : somme de contrôle tiger #R : p+i+n+u+g+s+m+c+md5 #L : p+i+n+u+g #E : Groupe vide #> : Journaux augmentant en tailles p+u+g+i+n+S # Courriel où les compte-rendus sont envoyés et le nombre de # lignes maximum. @@define MAILTO root @@define LINES 1000 # Quelques variables utiles pour les prisons chroot @@define CHRAPACHE /var/chroot/apache @@define CHRMYSQLD /var/chroot/mysql # Quelques règles prédéfinies Binlib = p+i+n+u+g+s+b+m+c+md5+sha1 ConfFiles = p+i+n+u+g+s+b+m+c+md5+sha1 Logs = p+i+n+u+g+S Devices = p+i+n+u+g+s+b+c+md5+sha1 Databases = p+n+u+g StaticDir = p+i+n+u+g ManPages = p+i+n+u+g+s+b+m+c+md5+sha1 StdTests = R+b+md5+sha1 </pre>
Suite . . .

Fichier de configuration de AIDE (suite)
<pre> # Enfin on choisit quels sont les répertoires et fichiers que nous # souhaitons mettre dans la base de donnée # Quelques répertoires importants du système # Pour le noyau =/boot\$ Binlib # Pour les binaires /bin Binlib /sbin Binlib /usr/bin Binlib /usr/sbin Binlib /usr/local/bin Binlib /usr/local/sbin Binlib /usr/games Binlib # Pour les documentations /usr/share/doc ManPages # Pour les répertoires des utilisateurs /home Binlib # Pour les bibliothèques /lib Binlib /usr/lib Binlib /usr/local/lib Binlib # Pour les journaux /var/log\$ StaticDir /var/log/aide/aide.log([0-9])?(.gz) ? Databases /var/log/aide/error.log([0-9])?(.gz) ? Databases /var/log/setuid.changes([0-9])?(.gz) ? Databases /var/log Logs # Pour les périphériques !/dev/pts /dev Devices # Autres fichiers et répertoires / StdTests /etc StdTests /var/spool/cron Databases /var/spool/cron/crontabs Databases /var/run\$ StaticDir !/var/run /proc\$ StaticDir !/proc # Quelques répertoires dont le contenu varie beaucoup /var/spool/squid\$ StaticDir !/var/spool/squid /home/arno/Mail\$ StaticDir !/home/arno/Mail /home/arno/.irssi\$ StaticDir !/home/arno/.irssi /home/arno/.imcom\$ StaticDir !/home/arno/.imcom ## Pour les environnements chroot </pre>
Suite . . .

Fichier de configuration de AIDE (suite)
<pre> ## Pour MySQL # Pour les binaires @@{CHRMYSQDL}/bin Binlib @@{CHRMYSQDL}/sbin Binlib @@{CHRMYSQDL}/usr/bin Binlib @@{CHRMYSQDL}/usr/sbin Binlib # Pour les bibliothèques @@{CHRMYSQDL}/lib Binlib @@{CHRMYSQDL}/usr/lib Binlib # Pour les journaux @@{CHRMYSQDL}/var/log\$ StaticDir @@{CHRMYSQDL}/var/log Logs # Autres fichiers @@{CHRMYSQDL}/var/run\$ StaticDir !@@{CHRMYSQDL}/var/run @@{CHRMYSQDL}/proc\$ StaticDir !@@{CHRMYSQDL}/proc # Les fichiers de configuration @@{CHRMYSQDL}/ StdTests @@{CHRMYSQDL}/etc StdTests ## Pour Apache # Pour Les binaires @@{CHRAPACHE}/bin Binlib @@{CHRAPACHE}/usr/sbin Binlib # Pour les bibliothèques @@{CHRAPACHE}/lib Binlib @@{CHRAPACHE}/usr/lib Binlib # Pour les journaux @@{CHRAPACHE}/var/log\$ StaticDir @@{CHRAPACHE}/var/log Logs # Pour les autres fichiers @@{CHRAPACHE}/var/run\$ StaticDir !@@{CHRAPACHE}/var/run @@{CHRAPACHE}/proc\$ StaticDir !@@{CHRAPACHE}/proc # Fichiers et répertoires de configuration @@{CHRAPACHE}/ StdTests @@{CHRAPACHE}/etc StdTests </pre>
Fin de /etc/aide/aide.conf

Maintenant, AIDE devrait se lancer de tous les jours via *Cron*. Vous pourrez ainsi lire ces rapports par l'intermédiaire de *mutt* ou, si vous n'avez pas installé ce dernier, avec la commande *mail*. Notez que les rapports seront envoyés à l'utilisateur 'toto' suivant la configuration de postfix que nous avons choisi (les courriels de l'utilisateur root sont automatiquement redirigés vers l'utilisateur toto).

Cependant, n'oubliez pas que si vous mettez à jour les paquets de votre système (cela devrait se produire assez rarement pour la version stable de Debian GNU/Linux et concernera très souvent des mises à jour de sécurité), AIDE vous dira que ces fichiers ont été modifiés, ce qui est parfaitement normal. Vous pourrez, une fois ces mises à jour et la vérification de la base de données effectuées, mettre à jour celle-ci en tapant tout simplement :

```
# aideinit
```

AIDE se révèle être selon moi un outil indispensable à l'administration d'un système, tout d'abord car il est puis-

sant et très simple à configurer qui plus est. On peut aisément le comparer à Tripwire, un logiciel équivalent mais pas tout à fait libre.

Pour plus d'informations au sujet de AIDE, vous pouvez consulter le manuel officiel [8] ou la page de manuel dédiée (`man aide`).

5.2 Logcheck

Logcheck est un utilitaire destiné à vérifier les journaux que vous souhaitez à la recherche d'événements suspects et envoie enfin ce rapport par courriel. Cela vous évite donc de consulter les journaux périodiquement et en entier, ce qui peut se révéler long et surtout peu pratique. De plus, vous pouvez définir le niveau d'alerte.

L'installation se fait très simplement pour Debian car Logcheck est disponible officiellement, on installe donc ce paquet :

```
# apt-get install "logcheck*"
```

Normalement quelques questions vous seront posées au sujet de la configuration de Logcheck. Choisissez l'adresse courriel vers laquelle seront envoyés les rapports générés par Logcheck. Ensuite répondez *<Yes>* à la question vous demandant si vous souhaitez créer un fichier comportant `/var/log/syslog` et qui va spécifier que ce journal va être examiné.

Lors de l'installation du paquet `logcheck-database`, choisissez le niveau de sécurité qui sera utilisé, le plus élevé étant *paranoid*, puis *server* et enfin *workstation*. Personnellement j'ai choisi le premier car il rapporte également les connexions effectuées depuis l'extérieur.

Désormais, Logcheck sera lancé toutes les heures via Cron et un courriel sera envoyé à l'utilisateur que vous avez choisi lors de l'installation. Vous pouvez modifier ces valeurs en tapant simplement la commande suivante pour logcheck :

```
# dpkg-reconfigure logcheck
```

Et enfin pour les options du paquet `logcheck-database` qui permet de régler le niveau de sécurité :

```
# dpkg-reconfigure logcheck-database
```

Vous pouvez également configurer Logcheck afin qu'il vérifie les journaux de votre choix. Cela se fait assez facilement :

Fichier /etc/logcheck/logcheck.logfiles
<i>## Voici les fichiers vérifiés par logcheck</i>
<i># Options par défaut</i>
<code>/var/log/messages</code>
<code>/var/log/syslog</code>
<code>/var/log/auth.log</code>
<code>/var/log/mail.log</code>
<code>/var/log/daemon.log</code>
<i>## Options pour les environnements chroot</i>
<i># Pour Apache</i>
<code>/var/chroot/apache/var/log/apache/access.log</code>
<code>/var/chroot/apache/var/log/apache/error.log</code>
<i># Pour MySQL</i>
<code>/var/chroot/mysqld/var/log/mysql.log</code>
<code>/var/chroot/mysqld/var/log/mysql/mysql.err</code>
Fin de /etc/logcheck/logcheck.logfiles

La configuration de cet outil est maintenant terminée, il faut bien avouer que celui-ci est très facile à configurer, je vous conseille donc vivement de l'utiliser.

6 Conclusion

Dans cette première partie de cet article, nous avons vu comment installer puis configurer quelques services couramment utilisés sur des serveurs. Nous avons également mis en place une passerelle et sécurisé de façon basique l'infrastructure. Dans une seconde partie, il sera expliqué comment configurer la passerelle pour un réseau Sans-Fil de la manière la plus sécurisée possible.

Références

- [1] « *Securing Debian Manual* » <http://www.debian.org/doc/manuals/securing-debian-howto/index.en.html>, document absolument indispensable pour la sécurisation d'un système sous Debian GNU/Linux. De plus il est disponible également en français.
- [2] « *Gérer les services lancés au démarrage* » <http://www.andesi.org/article.php?id=services>, article décrivant la gestion des services simplement sous Debian bien sûr.
- [3] « *Compiler son noyau Linux à la sauce Debian* » <http://www.andesi.org/article.php?id=noyau>, cet article explique comment compiler un noyau Linux à la sauce Debian. A lire si vous souhaitez mieux comprendre de quelle manière on compile un noyau.
- [4] « *The Linux Kernel Archives* » <http://www.kernel.org>, site sur lequel vous pourrez trouver la dernière version des sources du noyau Linux.
- [5] « *Security Information* » <http://www.debian.org/security/>, site officiel indiquant les dernières alertes de sécurité sur les paquets Debian.
- [6] *ext3-2.4-0.9.4* <http://lwn.net/2001/0802/a/ext3-modes.php3>, courriel de la liste de diffusion du noyau indiquant les différents modes disponibles pour ext3, en anglais mais intéressant quand même.
- [7] « *AIDE - Advanced Intrusion Detection Environment* » (<http://www.cs.tut.fi/~rammer/aide.html>), Site internet du projet AIDE où vous pourrez trouver une liste de diffusion ainsi que les sources bien sûr.
- [8] « *The Aide manual* » (<http://www.cs.tut.fi/~rammer/aide/manual.html>), le manuel officiel de AIDE. A lire si vous souhaitez en apprendre un peu plus sur cet outil.

7 Annexe

7.1 Arborescence d'une prison chroot pour Apache

Arborescence de la prison chroot pour Apache						
# ls -lRl /var/chroot/apache/						
/var/chroot/apache/ :						
total 32						
drwxr-xr-x	2 root	root	4096	déc 7 16 :04	bin	
drwxr-xr-x	2 root	root	4096	déc 7 16 :02	dev	
drwxr-xr-x	4 root	root	4096	déc 29 19 :24	etc	
drwxr-xr-x	2 root	root	4096	déc 7 16 :02	lib	
dr-xr-xr-x	2 root	root	4096	déc 7 15 :58	proc	
drwxrwxrwt	2 root	root	4096	jan 29 21 :18	tmp	
drwxr-xr-x	5 root	root	4096	déc 7 16 :02	usr	
drwxr-xr-x	5 root	root	4096	déc 29 22 :34	var	
/var/chroot/apache/bin :						
total 528						
-rwxr-xr-x	1 root	root	511400	déc 7 15 :58	bash	
-rwxr-xr-x	1 root	root	9308	déc 7 16 :04	cat	
-rwxr-xr-x	1 root	root	8956	déc 7 16 :03	false	
lrwxrwxrwx	1 root	root	4	déc 7 15 :58	sh -> bash	
/var/chroot/apache/dev :						
total 0						
-rw-r--r--	1 root	root	0	jan 29 21 :18	null	
crw-rw-rw-	1 root	tty	50	déc 7 15 :58	tty	
/var/chroot/apache/etc :						
total 60						
drwxr-xr-x	2 root	root	4096	déc 30 13 :16	apache	
-rw-r--r--	1 root	root	15	déc 7 16 :02	group	
-rw-r--r--	1 root	root	26	déc 7 16 :02	host.conf	
-rw-r--r--	1 root	root	289	déc 7 16 :02	hosts	
-rw-r--r--	1 root	root	1456	déc 29 19 :21	ld.so.cache	
lrwxrwxrwx	1 root	root	32	déc 7 16 :02	localtime	
\					-> /usr/share/zoneinfo/Europe/Paris	
-rw-r--r--	1 root	root	15723	déc 7 16 :02	mime.types	
-rw-r--r--	1 root	root	1982	déc 7 19 :00	my.cnf	
-rw-r--r--	1 root	root	465	déc 7 16 :02	nsswitch.conf	
-rw-r--r--	1 root	root	46	déc 7 16 :02	passwd	
drwxr-xr-x	3 root	root	4096	déc 7 16 :02	php4	
-rw-r--r--	1 root	root	1748	déc 7 16 :02	protocols	
-rw-r--r--	1 root	root	50	déc 7 16 :02	resolv.conf	
/var/chroot/apache/etc/apache :						
total 44						
-rw-r--r--	1 root	root	285	déc 7 16 :02	access.conf	
-rw-r--r--	1 root	root	34678	déc 30 13 :16	httpd.conf	
-rw-r--r--	1 root	root	297	déc 7 16 :02	srn.conf	
/var/chroot/apache/etc/php4 :						
total 4						
drwxr-xr-x	2 root	root	4096	déc 30 18 :45	apache	
/var/chroot/apache/etc/php4/apache :						
total 28						
Suite ...						

Arborescence de la prison chroot pour Apache (suite)						
-rw-r--r--	1	root	root	26856	déc 30 18 :45	php.ini
/var/chroot/apache/lib :						
total 2204						
-rwxr-xr-x	1	root	root	90210	déc 7 15 :58	ld-2.2.5.so
lrwxrwxrwx	1	root	root	11	déc 7 15 :58	ld-linux.so.2 -> ld-2.2.5.so
-rwxr-xr-x	1	root	root	1153784	déc 7 15 :58	libc-2.2.5.so
-rw-r--r--	1	root	root	19136	déc 7 15 :58	libcrypt-2.2.5.so
lrwxrwxrwx	1	root	root	17	déc 7 15 :58	libcrypt.so.1 -> libcrypt-2.2.5.so
lrwxrwxrwx	1	root	root	13	déc 7 15 :58	libc.so.6 -> libc-2.2.5.so
-rw-r--r--	1	root	root	49828	déc 7 15 :58	libdb1-2.2.5.so
lrwxrwxrwx	1	root	root	15	déc 7 15 :58	libdb2.so.2 -> libdb2.so.2.7.7
-rw-r--r--	1	root	root	262812	déc 7 15 :58	libdb2.so.2.7.7
lrwxrwxrwx	1	root	root	15	déc 7 15 :58	libdb.so.2 -> libdb1-2.2.5.so
-rw-r--r--	1	root	root	8008	déc 7 15 :58	libdl-2.2.5.so
lrwxrwxrwx	1	root	root	14	déc 7 15 :58	libdl.so.2 -> libdl-2.2.5.so
-rw-r--r--	1	root	root	130088	déc 7 15 :58	libm-2.2.5.so
lrwxrwxrwx	1	root	root	13	déc 7 15 :58	libm.so.6 -> libm-2.2.5.so
lrwxrwxrwx	1	root	root	17	déc 7 15 :58	libncurses.so.5 -> libncurses.so.5.2
-rw-r--r--	1	root	root	248132	déc 7 15 :58	libncurses.so.5.2
-rw-r--r--	1	root	root	69472	déc 7 16 :02	libnsl-2.2.5.so
lrwxrwxrwx	1	root	root	15	déc 7 16 :02	libnsl.so.1 -> libnsl-2.2.5.so
-rw-r--r--	1	root	root	40152	déc 7 16 :02	libnss_compat-2.2.5.so
lrwxrwxrwx	1	root	root	22	déc 7 16 :02	libnss_compat.so.2
\						-> libnss_compat-2.2.5.so
-rw-r--r--	1	root	root	12176	déc 7 16 :02	libnss_dns-2.2.5.so
lrwxrwxrwx	1	root	root	19	déc 7 16 :02	libnss_dns.so.2
\						-> libnss_dns-2.2.5.so
-rw-r--r--	1	root	root	32668	déc 7 16 :02	libnss_files-2.2.5.so
lrwxrwxrwx	1	root	root	21	déc 7 16 :02	libnss_files.so.2
\						-> libnss_files-2.2.5.so
lrwxrwxrwx	1	root	root	14	déc 7 16 :02	libpam.so.0 -> libpam.so.0.72
-rw-r--r--	1	root	root	29420	déc 7 16 :02	libpam.so.0.72
-rw-r--r--	1	root	root	56480	déc 7 16 :02	libresolv-2.2.5.so
lrwxrwxrwx	1	root	root	18	déc 7 16 :02	libresolv.so.2
\						-> libresolv-2.2.5.so
/var/chroot/apache/proc :						
total 0						
/var/chroot/apache/tmp :						
total 0						
-rw-----	1	root	root	0	jan 29 21 :18	session_mm_apache0.sem
/var/chroot/apache/usr :						
total 12						
drwxr-xr-x	4	root	root	4096	déc 29 19 :20	lib
drwxr-xr-x	2	root	root	4096	déc 7 15 :58	sbin
drwxr-xr-x	4	root	root	4096	déc 7 16 :14	share
/var/chroot/apache/usr/lib :						
total 540						
drwxr-xr-x	3	root	root	4096	déc 7 15 :58	apache
lrwxrwxrwx	1	root	root	15	déc 7 16 :02	libbz2.so.1.0 -> libbz2.so.1.0.2
-rw-r--r--	1	root	root	60392	déc 7 16 :02	libbz2.so.1.0.2
lrwxrwxrwx	1	root	root	17	déc 7 15 :58	libexpat.so.1 -> libexpat.so.1.0.0

Suite ...

Arborescence de la prison chroot pour Apache (suite)					
-rw-r--r--	1 root	root	131388	déc 7 15 :58	libexpat.so.1.0.0
lrwxrwxrwx	1 root	root	16	déc 7 16 :02	libmm.so.11 -> libmm.so.11.0.23
-rw-r--r--	1 root	root	14844	déc 7 16 :02	libmm.so.11.0.23
lrwxrwxrwx	1 root	root	24	déc 29 19 :20	libmysqlclient.so.10
\					-> libmysqlclient.so.10.0.0
-rw-r--r--	1 root	root	217348	sep 12 01 :11	libmysqlclient.so.10.0.0
lrwxrwxrwx	1 root	root	15	déc 7 16 :02	libpcre.so.3 -> libpcre.so.3.0.3
-rw-r--r--	1 root	root	34380	déc 7 16 :02	libpcre.so.3.0.3
lrwxrwxrwx	1 root	root	13	déc 7 16 :02	libz.so.1 -> libz.so.1.1.4
-rw-r--r--	1 root	root	55432	déc 7 16 :02	libz.so.1.1.4
drwxr-xr-x	3 root	root	4096	déc 7 15 :25	php4
/var/chroot/apache/usr/lib/apache :					
total 20					
drwxr-xr-x	2 root	root	4096	déc 29 19 :35	1.3
-rwxr-xr-x	1 root	root	14004	déc 7 15 :58	suexec
/var/chroot/apache/usr/lib/apache/1.3 :					
total 2204					
-rw-r--r--	1 root	root	222	déc 29 19 :35	000mod_vhost_alias.info
-rw-r--r--	1 root	root	135	déc 29 19 :35	010mod_env.info
-rw-r--r--	1 root	root	245	déc 29 19 :35	020mod_log_config.info
-rw-r--r--	1 root	root	209	déc 29 19 :35	030mod_log_agent.info
-rw-r--r--	1 root	root	236	déc 29 19 :35	030mod_log_referer.info
-rw-r--r--	1 root	root	151	déc 29 19 :35	040mod_mime_magic.info
-rw-r--r--	1 root	root	203	déc 29 19 :35	050mod_mime.info
-rw-r--r--	1 root	root	259	déc 29 19 :35	060mod_negotiation.info
-rw-r--r--	1 root	root	220	déc 29 19 :35	070mod_status.info
-rw-r--r--	1 root	root	189	déc 29 19 :35	080mod_info.info
-rw-r--r--	1 root	root	297	déc 29 19 :35	090mod_include.info
-rw-r--r--	1 root	root	312	déc 29 19 :35	100mod_autoindex.info
-rw-r--r--	1 root	root	126	déc 29 19 :35	110mod_dir.info
-rw-r--r--	1 root	root	225	déc 29 19 :35	120mod_cgi.info
-rw-r--r--	1 root	root	213	déc 29 19 :35	130mod_asis.info
-rw-r--r--	1 root	root	208	déc 29 19 :35	140mod_imap.info
-rw-r--r--	1 root	root	159	déc 29 19 :35	150mod_actions.info
-rw-r--r--	1 root	root	145	déc 29 19 :35	160mod_speling.info
-rw-r--r--	1 root	root	196	déc 29 19 :35	170mod_userdir.info
-rw-r--r--	1 root	root	242	déc 29 19 :35	190mod_alias.info
-rw-r--r--	1 root	root	297	déc 29 19 :35	200mod_rewrite.info
-rw-r--r--	1 root	root	143	déc 29 19 :35	210mod_access.info
-rw-r--r--	1 root	root	222	déc 29 19 :35	220mod_auth.info
-rw-r--r--	1 root	root	301	déc 29 19 :35	230mod_auth_anon.info
-rw-r--r--	1 root	root	250	déc 29 19 :35	240mod_auth_dbm.info
-rw-r--r--	1 root	root	242	déc 29 19 :35	250mod_auth_db.info
-rw-r--r--	1 root	root	366	déc 29 19 :35	260libproxy.info
-rw-r--r--	1 root	root	196	déc 29 19 :35	270mod_digest.info
-rw-r--r--	1 root	root	216	déc 29 19 :35	280mod_cern_meta.info
-rw-r--r--	1 root	root	169	déc 29 19 :35	290mod_expires.info
-rw-r--r--	1 root	root	130	déc 29 19 :35	300mod_headers.info
-rw-r--r--	1 root	root	151	déc 29 19 :35	310mod_usertrack.info
-rw-r--r--	1 root	root	114	déc 29 19 :35	320mod_unique_id.info
-rw-r--r--	1 root	root	184	déc 29 19 :35	330mod_setenvif.info
Suite ...					

Arborescence de la prison chroot pour Apache (suite)

-rw-r--r--	1	root	root	134	déc 29 19 :35	400mod_auth_sys.info
-rw-r--r--	1	root	root	151	déc 29 19 :35	400mod_put.info
-rw-r--r--	1	root	root	309	déc 29 19 :35	400mod_throttle.info
-rw-r--r--	1	root	root	156	déc 29 19 :35	500mod_allowdev.info
-rw-r--r--	1	root	root	197	déc 29 19 :35	500mod_eaccess.info
-rw-r--r--	1	root	root	188	déc 29 19 :35	500mod_php4.info
-rw-r--r--	1	root	root	166	déc 29 19 :35	500mod_roaming.info
-rw-r--r--	1	root	root	9096	déc 29 19 :35	libcache.so
-rw-r--r--	1	root	root	1271492	déc 29 19 :35	libphp4.so
-rw-r--r--	1	root	root	80772	déc 29 19 :35	libproxy.so
-rw-r--r--	1	root	root	7048	déc 29 19 :35	mod_access.so
-rw-r--r--	1	root	root	5804	déc 29 19 :35	mod_actions.so
-rw-r--r--	1	root	root	7956	déc 29 19 :35	mod_alias.so
-rw-r--r--	1	root	root	5680	déc 29 19 :35	mod_allowdev.so
-rw-r--r--	1	root	root	4656	déc 29 19 :35	mod_asis.so
-rw-r--r--	1	root	root	5772	déc 29 19 :35	mod_auth_anon.so
-rw-r--r--	1	root	root	11248	déc 29 19 :35	mod_auth_cache.so
-rw-r--r--	1	root	root	6896	déc 29 19 :35	mod_auth_cookie_file.so
-rw-r--r--	1	root	root	5684	déc 29 19 :35	mod_auth_cookie.so
-rw-r--r--	1	root	root	6640	déc 29 19 :35	mod_auth_dbm.so
-rw-r--r--	1	root	root	6536	déc 29 19 :35	mod_auth_db.so
-rw-r--r--	1	root	root	21140	déc 29 19 :35	mod_auth_digest.so
-rw-r--r--	1	root	root	8700	déc 29 19 :35	mod_auth_external.so
-rw-r--r--	1	root	root	4060	déc 29 19 :35	mod_auth_inst.so
-rw-r--r--	1	root	root	8912	déc 29 19 :35	mod_auth.so
-rw-r--r--	1	root	root	6016	déc 29 19 :35	mod_auth_sys.so
-rw-r--r--	1	root	root	5660	déc 29 19 :35	mod_auth_system.so
-rw-r--r--	1	root	root	26748	déc 29 19 :35	mod_autoindex.so
-rw-r--r--	1	root	root	14112	déc 29 19 :35	mod_bandwidth.so
-rw-r--r--	1	root	root	6460	déc 29 19 :35	mod_cern_meta.so
-rw-r--r--	1	root	root	11140	déc 29 19 :35	mod_cgi.so
-rw-r--r--	1	root	root	13000	déc 29 19 :35	mod_cgisock.so
-rw-r--r--	1	root	root	7592	déc 29 19 :35	mod_digest.so
-rw-r--r--	1	root	root	5660	déc 29 19 :35	mod_dir.so
-rw-r--r--	1	root	root	4680	déc 29 19 :35	mod_disallow_id.so
-rw-r--r--	1	root	root	19040	déc 29 19 :35	mod_eaccess.so
-rw-r--r--	1	root	root	5688	déc 29 19 :35	mod_env.so
-rw-r--r--	1	root	root	7032	déc 29 19 :35	mod_expires.so
-rw-r--r--	1	root	root	5656	déc 29 19 :35	mod_headers.so
-rw-r--r--	1	root	root	12144	déc 29 19 :35	mod_imap.so
-rw-r--r--	1	root	root	28484	déc 29 19 :35	mod_include.so
-rw-r--r--	1	root	root	15884	déc 29 19 :35	mod_info.so
-rw-r--r--	1	root	root	5756	déc 29 19 :35	mod_ip_forwarding.so
-rw-r--r--	1	root	root	5692	déc 29 19 :35	mod_lock.so
-rw-r--r--	1	root	root	4948	déc 29 19 :35	mod_log_agent.so
-rw-r--r--	1	root	root	12780	déc 29 19 :35	mod_log_config.so
-rw-r--r--	1	root	root	5704	déc 29 19 :35	mod_log_referer.so
-rw-r--r--	1	root	root	11452	déc 29 19 :35	mod_macro.so
-rw-r--r--	1	root	root	18488	déc 29 19 :35	mod_mime_magic.so
-rw-r--r--	1	root	root	11044	déc 29 19 :35	mod_mime.so
-rw-r--r--	1	root	root	21164	déc 29 19 :35	mod_negotiation.so

Suite ...

Arborescence de la prison chroot pour Apache (suite)						
-rw-r--r--	1	root	root	5752	déc 29 19 :35	mod_peekhole.so
-rw-r--r--	1	root	root	6076	déc 29 19 :35	mod_put.so
-rw-r--r--	1	root	root	4460	déc 29 19 :35	mod_qs2ssi.so
-rw-r--r--	1	root	root	44448	déc 29 19 :35	mod_rewrite.so
-rw-r--r--	1	root	root	10788	déc 29 19 :35	mod_roaming.so
-rw-r--r--	1	root	root	14340	déc 29 19 :35	mod_session.so
-rw-r--r--	1	root	root	7256	déc 29 19 :35	mod_setenvif.so
-rw-r--r--	1	root	root	8792	déc 29 19 :35	mod_speling.so
-rw-r--r--	1	root	root	15132	déc 29 19 :35	mod_status.so
-rw-r--r--	1	root	root	31972	déc 29 19 :35	mod_throttle.so
-rw-r--r--	1	root	root	6652	déc 29 19 :35	mod_ticket.so
-rw-r--r--	1	root	root	5724	déc 29 19 :35	mod_unique_id.so
-rw-r--r--	1	root	root	10452	déc 29 19 :35	mod_urlcount.so
-rw-r--r--	1	root	root	6080	déc 29 19 :35	mod_userdir.so
-rw-r--r--	1	root	root	8036	déc 29 19 :35	mod_usertrack.so
-rw-r--r--	1	root	root	6908	déc 29 19 :35	mod_vhost_alias.so
/var/chroot/apache/usr/lib/php4 :						
total 4						
drwxr-xr-x	2	root	root	4096	déc 29 19 :07	20010901
/var/chroot/apache/usr/lib/php4/20010901 :						
total 36						
-rw-r--r--	1	root	root	35716	jui 16 2003	mysql.so
/var/chroot/apache/usr/sbin :						
total 264						
-rwxr-xr-x	1	root	root	254216	éc 7 15 :58	apache
-rwxr-xr-x	1	root	root	7081	déc 7 15 :58	apachectl
/var/chroot/apache/usr/share :						
total 8						
drwxr-xr-x	3	root	root	4096	déc 7 15 :25	apache
drwxr-xr-x	3	root	root	4096	déc 7 16 :02	zoneinfo
/var/chroot/apache/usr/share/apache :						
total 4						
drwxr-xr-x	4	root	root	4096	déc 7 15 :25	icons
/var/chroot/apache/usr/share/zoneinfo :						
total 4						
drwxr-xr-x	2	root	root	4096	déc 7 16 :02	Europe
/var/chroot/apache/usr/share/zoneinfo/Europe :						
total 4						
-rw-r--r--	1	root	root	1082	déc 7 16 :02	Paris
/var/chroot/apache/var :						
total 12						
drwxr-xr-x	3	root	root	4096	déc 7 16 :00	log
drwxr-xr-x	2	root	root	4096	déc 7 19 :36	run
drwxr-xr-x	5	www-data	www-data	4096	déc 30 16 :31	www
/var/chroot/apache/var/log :						
total 4						
drwxr-xr-x	2	root	root	4096	jan 25 06 :31	apache
/var/chroot/apache/var/log/apache :						
total 152						
-rw-r--r--	1	root	adm	16491	jan 31 00 :16	access.log
-rw-r--r--	1	root	adm	2166	jan 31 00 :16	error.log
Suite ...						

Arborescence de la prison chroot pour Apache (suite)						
/var/chroot/apache/var/run :						
total 4						
-rw-r--r--	1	root	root	5	jan 29 21 :18	apache.pid
/var/chroot/apache/var/www :						
total 20						
-rw-r--r--	1	www-data	www-data	3063	déc 13 18 :45	index.html
Fin de l'arborescence de la prison chroot de Apache						

7.2 Arborescence d'une prison chroot pour MySQL

Arborescence de la prison chroot pour MySQL						
# ls -lRl /var/chroot/mysqld/						
/var/chroot/mysqld/ :						
total 40						
drwxr-xr-x	2	root	root	4096	déc 7 18 :54	bin
drwxr-xr-x	2	root	root	4096	déc 7 18 :49	dev
drwxr-xr-x	3	root	root	4096	déc 7 18 :58	etc
drwxr-xr-x	2	root	root	4096	déc 7 18 :54	lib
dr-xr-xr-x	2	root	root	4096	déc 7 18 :54	proc
drwxr-xr-x	2	root	root	4096	déc 7 18 :55	sbin
drwxrwxrwx	2	root	root	4096	déc 7 18 :59	tmp
drwxr-xr-x	6	root	root	4096	déc 7 18 :49	usr
drwxr-xr-x	5	root	root	4096	déc 7 18 :55	var
/var/chroot/mysqld/bin :						
total 652						
-rwxr-xr-x	1	root	root	511400	déc 7 18 :49	bash
-rwxr-xr-x	1	root	root	17896	déc 7 18 :49	chown
-rwxr-xr-x	1	root	root	25820	déc 7 18 :49	date
-rwxr-xr-x	1	root	root	10844	déc 7 18 :54	echo
-rwxr-xr-x	1	root	root	9784	déc 7 18 :49	hostname
-rwxr-xr-x	1	root	root	25352	déc 7 18 :49	rm
-rwxr-xr-x	1	root	root	21132	déc 7 18 :54	sed
lrwxrwxrwx	1	root	root	4	déc 7 18 :49	sh -> bash
-rwxr-xr-x	1	root	root	23176	déc 7 18 :49	touch
/var/chroot/mysqld/dev :						
total 4						
-rw-rw---	1	root	root	59	jan 29 18 :47	null
crw-rw-rw-	1	root	tty	50	déc 7 18 :49	tty
/var/chroot/mysqld/etc :						
total 20						
-rw-r--r--	1	root	root	13	déc 7 18 :54	group
-rw-r--r--	1	root	root	289	déc 7 15 :02	hosts
lrwxrwxrwx	1	root	root	32	déc 7 18 :49	localtime -> /usr/share/zoneinfo/Europe/Paris
drwxr-xr-x	2	root	root	4096	déc 7 18 :49	mysql
-rw-r--r--	1	root	root	465	déc 7 18 :54	nsswitch.conf
-rw-r--r--	1	root	root	55	déc 7 18 :54	passwd
/var/chroot/mysqld/etc/mysql :						
total 4						
-rw-r--r--	1	root	root	1964	déc 7 18 :49	my.cnf
/var/chroot/mysqld/lib :						
Suite ...						

Arborescence de la prison chroot pour MySQL (suite)						
total 1968						
-rwxr-xr-x	1	root	root	90210	déc 7 18 :49	ld-2.2.5.so
lrwxrwxrwx	1	root	root	11	déc 7 18 :49	ld-linux.so.2 -> ld-2.2.5.so
-rwxr-xr-x	1	root	root	1153784	déc 7 18 :49	libc-2.2.5.so
-rw-r--r--	1	root	root	19136	déc 7 18 :49	libcrypt-2.2.5.so
lrwxrwxrwx	1	root	root	17	déc 7 18 :49	libcrypt.so.1 -> libcrypt-2.2.5.so
lrwxrwxrwx	1	root	root	13	déc 7 18 :49	libc.so.6 -> libc-2.2.5.so
-rw-r--r--	1	root	root	8008	déc 7 18 :49	libdl-2.2.5.so
lrwxrwxrwx	1	root	root	14	déc 7 18 :49	libdl.so.2 -> libdl-2.2.5.so
-rw-r--r--	1	root	root	130088	déc 7 18 :49	libm-2.2.5.so
lrwxrwxrwx	1	root	root	13	déc 7 18 :49	libm.so.6 -> libm-2.2.5.so
lrwxrwxrwx	1	root	root	17	déc 7 18 :49	libncurses.so.5 -> libncurses.so.5.2
-rw-r--r--	1	root	root	248132	déc 7 18 :49	libncurses.so.5.2
-rw-r--r--	1	root	root	69472	déc 7 18 :49	libnsl-2.2.5.so
lrwxrwxrwx	1	root	root	15	déc 7 18 :49	libnsl.so.1 -> libnsl-2.2.5.so
-rw-r--r--	1	root	root	40152	déc 7 18 :54	libnss_compat-2.2.5.so
lrwxrwxrwx	1	root	root	22	déc 7 18 :54	libnss_compat.so.2 -> libnss_compat-2.2.5.so
-rw-r--r--	1	root	root	32668	déc 7 18 :54	libnss_files-2.2.5.so
lrwxrwxrwx	1	root	root	21	déc 7 18 :54	libnss_files.so.2 -> libnss_files-2.2.5.so
-rw-r--r--	1	root	root	102172	déc 7 18 :49	libpthread-0.9.so
lrwxrwxrwx	1	root	root	17	déc 7 18 :49	libpthread.so.0 -> libpthread-0.9.so
-rw-r--r--	1	root	root	56480	déc 7 18 :49	libresolv-2.2.5.so
lrwxrwxrwx	1	root	root	18	déc 7 18 :49	libresolv.so.2 -> libresolv-2.2.5.so
lrwxrwxrwx	1	root	root	16	déc 7 18 :49	libwrap.so.0 -> libwrap.so.0.7.6
-rw-r--r--	1	root	root	24328	déc 7 18 :49	libwrap.so.0.7.6
/var/chroot/mysql/proc :						
total 0						
/var/chroot/mysql/sbin :						
total 0						
/var/chroot/mysql/tmp :						
total 0						
/var/chroot/mysql/usr :						
total 16						
drwxr-xr-x	2	root	root	4096	déc 7 18 :57	bin
drwxr-xr-x	2	root	root	4096	déc 7 18 :49	lib
drwxr-xr-x	2	root	root	4096	déc 7 18 :49	sbin
drwxr-xr-x	4	root	root	4096	déc 7 18 :54	share
/var/chroot/mysql/usr/bin :						
total 236						
-rwxr-xr-x	1	root	root	155016	déc 7 18 :49	my_print_defaults
-rwxr-xr-x	1	root	root	22696	déc 7 18 :56	mysqladmin
-rwxr-xr-x	1	root	root	12912	déc 7 18 :57	mysql_install_db
-rwxr-xr-x	1	root	root	11932	déc 7 18 :49	nice
-rwxr-xr-x	1	root	root	2500	déc 7 18 :49	nohup
-rwxr-xr-x	1	root	root	8406	déc 7 18 :49	safe_mysql
-rwxr-xr-x	1	root	root	10844	déc 7 18 :49	tee
lrwxrwxrwx	1	root	root	10	déc 7 18 :49	touch -> /bin/touch
/var/chroot/mysql/usr/lib :						
total 348						
-rw-r--r--	1	root	root	288540	déc 7 18 :49	libstdc++-3-libc6.2-2-2.10.0.so
lrwxrwxrwx	1	root	root	31	déc 7 18 :49	libstdc++-libc6.2-2.so.3 \
Suite . . .						

Arborescence de la prison chroot pour MySQL (suite)						
						-> libstdc++-3-libc6.2-2-2.10.0.so
lrwxrwxrwx	1	root	root	13	déc 7 18 :49	libz.so.1 -> libz.so.1.1.4
-rw-r--r--	1	root	root	55432	déc 7 18 :49	libz.so.1.1.4
/var/chroot/mysqld/usr/sbin :						
total 3452						
-rwxr-xr-x	1	root	root	3527080	déc 7 18 :49	mysqld
/var/chroot/mysqld/usr/share :						
total 8						
drwxr-xr-x	4	mysql	root	4096	déc 7 18 :54	mysql
drwxr-xr-x	3	root	root	4096	déc 7 18 :49	zoneinfo
/var/chroot/mysqld/usr/share/mysql :						
total 8						
drwxr-xr-x	2	mysql	root	4096	déc 7 18 :54	charsets
drwxr-xr-x	2	mysql	root	4096	déc 7 18 :54	english
/var/chroot/mysqld/usr/share/mysql/charsets :						
total 4						
-rw-r--r--	1	mysql	root	549	déc 7 18 :54	Index
/var/chroot/mysqld/usr/share/mysql/english :						
total 12						
-rw-r--r--	1	mysql	root	11158	déc 7 18 :54	errmsg.sys
/var/chroot/mysqld/usr/share/zoneinfo :						
total 4						
drwxr-xr-x	2	root	root	4096	déc 7 18 :49	Europe
/var/chroot/mysqld/usr/share/zoneinfo/Europe :						
total 4						
-rw-r--r--	1	root	root	1082	déc 7 18 :49	Paris
/var/chroot/mysqld/var :						
total 12						
drwxr-xr-x	3	root	root	4096	déc 7 18 :49	lib
drwxr-xr-x	3	root	root	4096	jan 30 06 :33	log
drwxr-xr-x	3	root	root	4096	déc 7 18 :55	run
/var/chroot/mysqld/var/lib :						
total 4						
drwxr-xr-x	6	mysql	mysql	4096	déc 30 17 :06	mysql
/var/chroot/mysqld/var/lib/mysql :						
total 24						
-rw-r--r--	1	root	root	48	déc 7 15 :31	my.cnf
drwxr-xr-x	2	mysql	root	4096	déc 7 15 :31	mysql
drwxr-xr-x	2	mysql	root	4096	déc 7 15 :31	test
/var/chroot/mysqld/var/lib/mysql/test :						
total 0						
/var/chroot/mysqld/var/log :						
total 36						
drwxrwx---	2	mysql	mysql	4096	jan 30 06 :33	mysql
-rw-----	1	mysql	mysql	155	jan 31 02 :32	mysql.log
/var/chroot/mysqld/var/log/mysql :						
total 32						
-rw-----	1	root	root	109	jan 31 02 :32	mysql.err
/var/chroot/mysqld/var/run :						
total 4						
drwxr-xr-x	2	mysql	root	4096	jan 31 02 :32	mysqld
Suite . . .						

Arborescence de la prison chroot pour MySQL (suite)						
/var/chroot/mysqld/var/run/mysqld :						
total 4						
-rw-rw---	1	mysql	mysql	5	jan 31 02 :32	mysqld.pid
srwxrwxrwx	1	mysql	mysql	0	jan 31 02 :32	mysqld.sock
Fin de l'arborescence de la prison chroot de MySQL						